
▼ Programming Foundation

- variable
- data type
- data structure
- function
- control flow

▼ Basic Calculation

```
1 1+1
```

```
2
```

```
1 5-2
```

```
3
```

```
1 5*2
```

```
10
```

```
1 7/2
```

```
3.5
```

```
1 5**2 # 5*5
```

```
25
```

```
1 # comment
2 # note to this code cell
3 print("hello world")
```

```
[1] "hello world"
```

```
1
```

```
1 1 + 2
```

▼ Variables

```
1 income <- 35000
2 expense <- 25000
3
4 saving <- income - expense
5
6 print(saving)
```

```
[1] 10000
```

```
1 print(income)
2 print(expense)
3 print(saving)
```

```
[1] 35000
[1] 25000
[1] 10000
```

```
1 # create new variable
2 my_name <- "Beckham"
3 print(my_name)
```

```
[1] "Beckham"
```

```
1 # remove variable
2 rm(my_name)
```

```
1 print(my_name)
```

```
Error in eval(expr, envir, enclos): object 'my_name' not found
Traceback:
```

```
1. print(my_name)
```

SEARCH STACK OVERFLOW

```
1 school <- "DataRockie"
2 print(school)
```

```
[1] "DataRockie"
```

```
1 # save income for 10 years, interest rate 5%
2
3 100000 * (1+0.05) ** 20
```

```
265329.770514442
```

```
1
```

▼ Data Types

- numeric
- character (text, string)
- logical (TRUE, FALSE)
- date
- factor (categorical - stats, special in R)

```
1 # numeric
2 gpa <- 3.41
3 print(gpa)
```

```
[1] 3.41
```

```
1 age <- 35
2 print(age)
3 print(class(age))
```

```
[1] 35
[1] "numeric"
```

```
1 class(gpa)
```

```
'numeric'
```

```
1 ## character
2 class("hello world")
```

```
'character'
```

```
1 ## concat two character
2 paste0("hello", "world")
```

```
.. ..  
1 paste("hello", "world", "hahaha")
```

```
'hello world hahaha'
```

```
1 # boolean => logical (TRUE, FALSE)
```

```
1 1+1 == 2
```

```
TRUE
```

```
1 # not equal
```

```
2 5 != 5
```

```
FALSE
```

```
1 !(1+1 == 2)
```

```
FALSE
```

```
1 # is R case sensitive?
```

```
2 "hi" == "Hi"
```

```
FALSE
```

```
1 my_name = "toy"
```

```
2 my_Name = "top"
```

```
1 my_name == my_Name
```

```
FALSE
```

```
1 ## date
```

```
2 today_date <- as.Date("2023-11-11")
```

```
3 print(today_date)
```

```
4 print(class(today_date))
```

```
[1] "2023-11-11"
```

```
[1] "Date"
```

```
1 as.character(today_date)
```

```
'2023-11-11'
```

```
1 ## as.____() change data type
```

```
2 as.integer(FALSE)
```

```
0
```

```
1 as.character(1234)
```

```
'1234'
```

```
1 class(123)
```

```
'numeric'
```

```
1 ## R name come from?
```

```
2 ## country of origin
```

```
3
```

```
4 ## Ross Ihaka + Robert Gentleman
```

```
5
```

```
6 ## vector => similar to Google Sheets Array
```

```
7 gender <- c("male", "female", "male")
```

```
8
```

```
9 gender <- factor(gender)
```

```
1 class(gender)
```

```
'factor'
```

```
1 ## factor has two categories
```

```
2 ## 1. nominal: male, female
```

```
3 ## 2. ordinal: low < med < high
```

```
1
```

▼ Data Structures

- vector (single data type)
- matrix (single data type)
- list
- dataframe

```
1 ## vector
```

```
2 scores <- c(88, 90, 50, 65, 76)
```

```
1 ## vectorization (no need to write loop)
```

```
2 scores + 2
```

```
90 + 92 + 52 + 67 + 78
```

```
1 ## sum scores, average score
```

```
2 ## aggregate functions
```

```
3 print(sum(scores))
```

```
4 print(mean(scores))
```

```
5 print(min(scores))
```

```
6 print(max(scores))
```

```
7 print(length(scores)) ## count
```

```
[1] 369
```

```
[1] 73.8
```

```
[1] 50
```

```
[1] 90
```

```
[1] 5
```

```
1 ## subset
```

```
2 scores <- c(
```

```
3   john = 88,
```

```
4   minnie = 90,
```

```
5   david = 50,
```

```
6   marry = 65,
```

```
7   anna = 76)
```

```
8
```

```
9 scores
```

```
john:      88 minnie:      90 david:      50 marry:      65 anna:      76
```

```
1 ## index starts at 1
```

```
2 ## by position 1,2,3
```

```
3 ## by name "david", "marry"
```

```
4 ## by condition
```

```
5
```

```
6 scores[scores > 80]
```

```
john:      88 minnie:      90
```

```
1 ## update value in scores
```

```
2 scores[3] <- 60
```

```
3 scores
```

```
john:      88 minnie:      90 david:      60 marry:      65 anna:      76
```

```
1 scores[5] <- 86
```

```
2 scores
```

john: 88 minnie: 90 david: 60 marry: 65 anna: 86

```
1 ## friends 5 persons
2 friends <- c("toy", "john", "mary", "david", "lisa")
```

```
1 paste("hi", friends)

'hi toy' · 'hi john' · 'hi mary' · 'hi david' · 'hi lisa'
```

```
1 ## matrix => vector 2 dimensions
2 m1 <- matrix(1:100, nrow=10, byrow=TRUE)
3 m1 * 2 # vectorization
```

A matrix: 10 × 10 of type dbl

2	4	6	8	10	12	14	16	18	20
22	24	26	28	30	32	34	36	38	40
42	44	46	48	50	52	54	56	58	60
62	64	66	68	70	72	74	76	78	80
82	84	86	88	90	92	94	96	98	100
102	104	106	108	110	112	114	116	118	120
122	124	126	128	130	132	134	136	138	140
142	144	146	148	150	152	154	156	158	160
162	164	166	168	170	172	174	176	178	180
182	184	186	188	190	192	194	196	198	200

```
1 ## matrix multiplication .dot
2 m1 <- matrix(c(2,4,5,10), ncol=2)
3
4 m2 <- matrix(c(1,1,5,2), ncol=2)
5
6 m1; m2
```

```
A
matrix:
2 × 2 of
type dbl
  2    5
```

```
1 # dot operation
2 m1 %*% m2
```

```
A matrix:
2 × 2 of
type dbl
  7    20
14   40
```

```
1 # dimension
2 dim(m1)
```

```
2 · 2
```

```
1 ## list , dataframe
2 ## list (key = value)
3 customer01 <- list(
4   name = "toy",
5   age = 35,
6   city = "Bangkok",
7   favorite_films = c("Dark Knight", "The Marvels")
8 )
9
10 customer02 <- list(
11   name = "jane",
12   age = 28,
13   city = "Seoul",
14   favorite_films = c("About Time", "Taylor Swift Concert")
15 )
16
17 # nested lists
18 list_customers <- list(customer01, customer02)
```

```
1 customer[["age"]] ## return value
```

```
35
```

```
1 customer["age"] ## return list
```

```
$age = 35
```



```
1 customer[["favorite_films"]][2]
```

```
'The Marvels'
```

```
1 customer[["city"]]
```

```
'Bangkok'
```

```
1 cust_name = list_customers[[2]][["name"]]
```

```
2 print(cust_name)
```

```
[1] "jane"
```

```
1 # dataframe
```

```
2 # the most important in data analyst role
```

```
3
```

```
4 # create a new dataframe from scratch
```

```
5
```

```
6 df <- data.frame(
```

```
7   id = 1:5,
```

```
8   name = c("toy", "john", "mary", "jane", "anne"),
```

```
9   age = c(28, 30, 31, 22, 25),
```

```
10  movie_lover = c(T, T, F, F, T)
```

```
11 )
```

```
1 # by condition
```

```
2 # &=and |=or
```

```
3 df2 <- df[ (df$age >= 30) | (df$movie_lover) , ]
```

```
1 ## create city ( a new column )
```

```
2 df$city <- c(rep("BKK",3), rep("LONDON",2)) # replicate
```

```
3 df
```

A data.frame: 5 × 5

	id	name	age	movie_lover	city
	<int>	<chr>	<dbl>	<lgl>	<chr>
	1	toy	28	TRUE	BKK
	2	john	30	TRUE	BKK
	3	mary	31	FALSE	BKK
	4	jane	22	FALSE	LONDON
	5	anne	25	TRUE	LONDON

```

1 df$random <- 100
2
3 ## remove column
4 df$random <- NULL
5 df

```

```

      A data.frame: 5 × 5
   id  name  age movie_lover  city
<int> <chr> <dbl>      <lgl> <chr>
1     1  toy   28        TRUE   BKK
2     2 john   30        TRUE   BKK
3     3 mary   31       FALSE   BKK
4     4 jane   22       FALSE LONDON
5     5 anne   25        TRUE LONDON

```

```

1 ## write csv file
2 write.csv(df, "movie.csv", row.names=FALSE)

```

```

1 ## read csv file
2 movie <- read.csv("movie.csv")
3 movie

```

```

      A data.frame: 5 × 5
   id  name  age movie_lover  city
<int> <chr> <int>      <lgl> <chr>
1     1  toy   28        TRUE   BKK
2     2 john   30        TRUE   BKK
3     3 mary   31       FALSE   BKK
4     4 jane   22       FALSE LONDON
5     5 anne   25        TRUE LONDON

```

```

1 ## structure
2 str(movie)

```

```

'data.frame':   5 obs. of  5 variables:
 $ id          : int  1 2 3 4 5
 $ name        : chr  "toy" "john" "mary" "jane" ...
 $ age         : int  28 30 31 22 25
 $ movie_lover : logi  TRUE TRUE FALSE FALSE TRUE
 $ city        : chr  "BKK" "BKK" "BKK" "LONDON" ...

```

▼ Function

```
1 sum( 1:10 )
```

```
55
```

```
1 ## create new function
2 my_secret_formula <- function(start, end) {
3   (start + end) * end / 2
4 }
5
6 my_secret_formula(1, 10)
```

```
55
```

```
1 add_two_num <- function(x,y) x+y
```

```
1 result <- add_two_num(5,4)
2 result
```

```
9
```

```
1 greeting <- function() print("hi!")
2 greeting()
```

```
[1] "hi!"
```

```
1 greeting <- function(name = "John Wick") {
2   return( paste("Hi!", name) )
3 }
4
5 greeting()
```

```
'Hi! John Wick'
```

```
1 greeting_city <- function(name="John", city="London") {
2   paste("Hi!", name, "Welcome to", city)
3 }
```

```
1 greeting_city(city="Bangkok", name="Mary")
```

```
'Hi! Mary Welcome to Bangkok'
```

```
1 greeting_city()
```

```
'Hi! John Welcome to London'
```

▼ function take user input

```
1 username <- readline("What is your name: ")
2 password <- as.numeric(readline("Password: "))
```

```
What is your name: toy
Password: 123456
```

```
1 class(username)
2 class(password)
```

```
'character'
'integer'
```

```
1 greeting_v2 <- function() {
2   ## readline get input in character
3   user_name = readline("Hi what's your name: ")
4   text = paste("Good morning!", user_name)
5   return(text)
6 }
```

```
1 greeting_v2()
```

```
Hi what's your name: John Wick
'Good morning! John Wick'
```

▼ Control Flow

- if
- for loop
- while loop

```
1 ## ifelse
2 ## =IF() in google sheets
3 ## ifelse() in R
4
5 score <- 79
6
7 ifelse(score >= 80, "passed","failed")
```

```
'failed'
```

```

1 ## if-else
2 grading <- function(score) {
3   if ( score >= 80 ) {
4     return("A")
5   } else if (score >= 70) {
6     return("B")
7   } else if (score >= 60) {
8     return("C")
9   } else if (score >= 50) {
10    return("D")
11  } else {
12    return("Failed")
13  }
14 }

```

```

1 ## function is re-usable
2 grading(57)

```

```
'D'
```

```

1 ## for loop
2 ## ไม่จำเป็นต้องใช้เท่าไรใน R
3 ## vectorization
4
5 fruits <- c("banana", "apple", "pineapple")

```

```

1 for (fruit in fruits) {
2   print(toupper(fruit) )
3 }

```

```

[1] "BANANA"
[1] "APPLE"
[1] "PINEAPPLE"

```

```
1 toupper(fruits)
```

```
'BANANA' · 'APPLE' · 'PINEAPPLE'
```

```

1 # while loop
2 count <- 0
3
4 while (count < 5) {
5   print("hi!")
6   count = count + 1
7 }

```

```

[1] "hi!"
[1] "hi!"
[1] "hi!"

```

```
[1] "hi!"  
[1] "hi!"
```

```
1 apply(mtcars, 2, mean)
```

```
mpg:      20.090625 cyl:      6.1875 disp:      230.721875 hp:      146.6875 drat:  
      3.5965625 wt:      3.21725 qsec:      17.84875 vs:      0.4375 am:      0.40625 gear:  
      3.6875 carb:      2.8125
```

Summary Basic Programming

1. variable
2. data type: numeric, character, logical, date, factor
3. data structure: vector matrix list dataframe
4. function
5. control flow: if for while

Homework

1. game() ==> เป่าขลุ่ย
2. chatbot() ==> user สั่งพิซซ่าได้